化工应用数学 第三章 数据处理 讲义

（04.最小二乘拟合）

数据/曲线拟合：

在工程实际或科学实验过程中，有时我们的主要目的是寻找相关参数之间的内在规律，即利用测量得到的离散数据群(x_i,y_i)，确定数据之间的经验的或者半经验的数学模型。由于事先已经有了经验行的数学模型，所以这一过程的主要任务就是通过数据去拟合得到数学模型中的参数。

数据拟合具有以下的特点：

1. 数据都有误差：由于测量误差的存在，数据不可避免的存在误差，所以在拟合过程中我们不能强制拟合得到的曲线精确的通过各点（例如在插值函数中所遇到的情况）

2. 数据量很大：在拟合过程中，我们已知的数据量往往会很大，数据量会远大于我们需要确定的经验关联式中的参数个数

曲线拟合是指求一条曲线，使得数据点均在此曲线的上方或下方不远处，所求的曲线叫做拟合曲线。拟合曲线能够保证局部没有较大波动的前提下，反应数据的总体分布，同时也能反应被逼近函数的特性，使求得的逼近函数与已知函数的偏差按某种特定方法度量达到最小。而这种度量方法我们一般采用的是最小二乘法。

注意：与插值问题不同，曲线拟合并不要求拟合曲线通过所有已知数据点，而是要求得到的近似函数能反应数据的基本关系。

实例 1.长度测量：

实际上，我们在日常生活或者之前做实验的时候都已经接触过了最小二乘法的使用，只是大家可能没有注意到，比如我们对于长度测量。

在测量长度的时候，有时候为了更加的精确，我们会多读取几次数据、或者使用不同的尺子测量长度，然后计算平均值（算数平均值）去作为测量的最终结果。我们会认为这样测量出的结果更加的精确，但是为什么呢？另外为什么是算数平均数而不是几何平均数/中位数/调和平均数呢？

这里实际上我们就是使用了最小二乘法的思想。

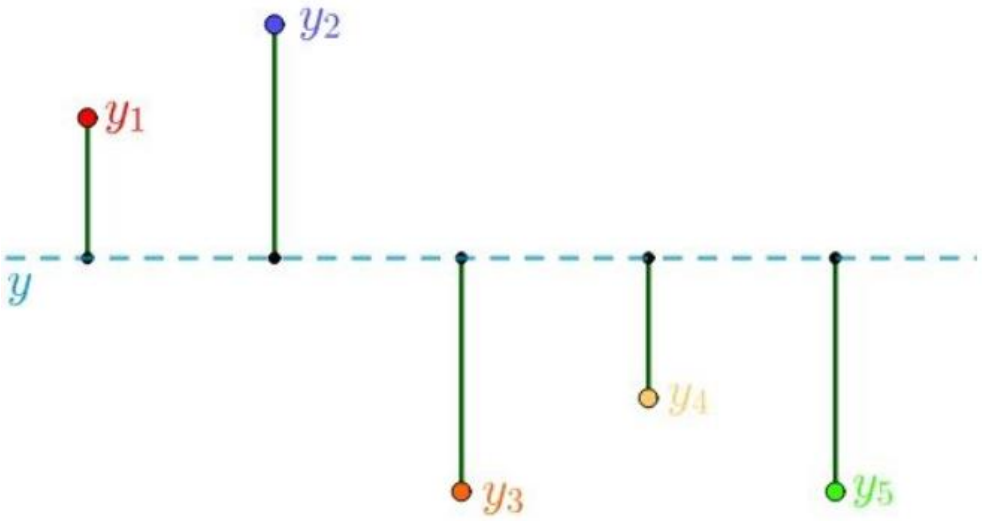
针对这个问题，实际数据只有一个真值 y，从直觉出发，如果误差是随机的，那么测量值应该是围绕真值上下波动，同时为了避免出现负数与去绝对值较复杂，所以法国数学家阿德里安·马里·勒让德提出让误差的平方和最小的 y 值就是真值，其中测量得到的数据与真值之间误差的平方和可以表述为：

$$S_{\epsilon^2} = \sum (y - y_i)^2$$

根据微积分的知识，上式在其对 y 求导等于 0 时有极小值，即：

$$\frac{d}{dy}S_{\epsilon^2} = \frac{d}{dy}\sum(y-y_i)^2 = 2\sum(y-y_i)$$

$$= 2((y-y_1)+(y-y_2)+(y-y_3)+(y-y_4)+(y-y_5)) = 0$$



整理上式我们可以得到：

$$5y = y_1 + y_2 + y_3 + y_4 + y_5 \implies y = \frac{y_1 + y_2 + y_3 + y_4 + y_5}{5}$$

可以发现，这里我们经验上的取算数平均值与最小二乘法是一致的。

最小二乘法：

最小二乘法（又称最小平方法）是一种数学优化方法。它通过最小化误差的平方和寻找数据的最佳函数匹配，即使得求得的数据与实际数据之间误差的平方和最小。

比如，我们假设需要拟合的函数为：

$$\varphi^*(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + a_n\varphi_n(x) = \sum_{k=0}^{n} a_k\varphi_k(x)$$

根据最小二乘法，最优化的参数选择是能满足：

$$S(a_0, a_1, \cdots, a_n) = \sum_{i=1}^{m}[a_0\varphi_0(x_i) + a_1\varphi_1(x_i) + a_n\varphi_n(x_i) - y_i]^2 = \min$$

根据高等数学的知识，函数 S 对各参数 ai 的偏导数等于 0 即为满足上式要求的参数，即

$$\frac{\partial S}{\partial a_k} = 0 \text{ 或 } \sum_{i=1}^{m}\varphi_k(x_i)[a_0\varphi_0(x_i) + a_1\varphi_1(x_i) + a_n\varphi_n(x_i) - y_i] = 0(k = 0,1,\cdots,n)$$

为什么是二乘(误差平方和)-高斯：

补充知识 1：误差是符合正态分布的(https://songshuhui.net/archives/76501)：

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

补充知识 2：极大似然法

设真值为θ，而$x_1,x_2,...,x_n$为 n 次独立测量值，每次测量的误差为，假设误差e_i的密度函数为 f(e)，则测量值的联合概率为 n 个误差的联合概率，记为：

$$L(\theta)=L(\theta;x_1,x_2,...,x_n)=\prod_{i=1}^{n}f(e_i)=\prod_{i=1}^{n}f(x_i-\theta) \tag{21}$$

但是高斯不采用贝叶斯的推理方式，而是直接取$L(\theta)$达到最大值的$\hat{\theta}=\hat{\theta}(x_1,...,x_n)$作为$\theta$的估计值，即

$$\hat{\theta}=\arg\max_{\theta}L(\theta) \tag{22}$$

现在我们把$L(\theta)$称为样本的似然函数，而得到的估计值$\hat{\theta}$称为极大似然估计。高斯首次给出了极大似然的思想，这个思想后来被统计学家 R.A.Fisher 系统的发展成为参数估计中的极大似然估计理论。

记样本为$(x^{(i)},y^{(i)})$，对样本的预测为$\hat{y}^{(i)}|_{\boldsymbol{\theta}}$该记法表示该预测依赖于参数$\boldsymbol{\theta}$的选取。我们有：

$$\boldsymbol{y}=\hat{\boldsymbol{y}}|_{\boldsymbol{\theta}}+\boldsymbol{\epsilon}$$

其中，$\boldsymbol{\epsilon}$是一个误差函数，我们通常认为其服从正态分布即

$$\boldsymbol{\epsilon}\sim N(0,\sigma^2)$$

因此有

$$\begin{aligned}\boldsymbol{y}-\hat{\boldsymbol{y}}|_{\boldsymbol{\theta}}&\sim N(0,\sigma^2)\\ \boldsymbol{y}&\sim N(\hat{\boldsymbol{y}}|_{\boldsymbol{\theta}},\sigma^2)\end{aligned}$$

要求$\boldsymbol{\theta}$的极大似然估计，即是说，我们现在得到的这个真实存在的$\boldsymbol{y}$在$\boldsymbol{\theta}$不同的取值下，出现概率最大，我们来看这个概率。令

$$L(\boldsymbol{\theta})=P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})=\prod_{i=1}^{m}\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(y^{(i)}-\hat{y}^{(i)}|_{\boldsymbol{\theta}})^2}{2\sigma})$$

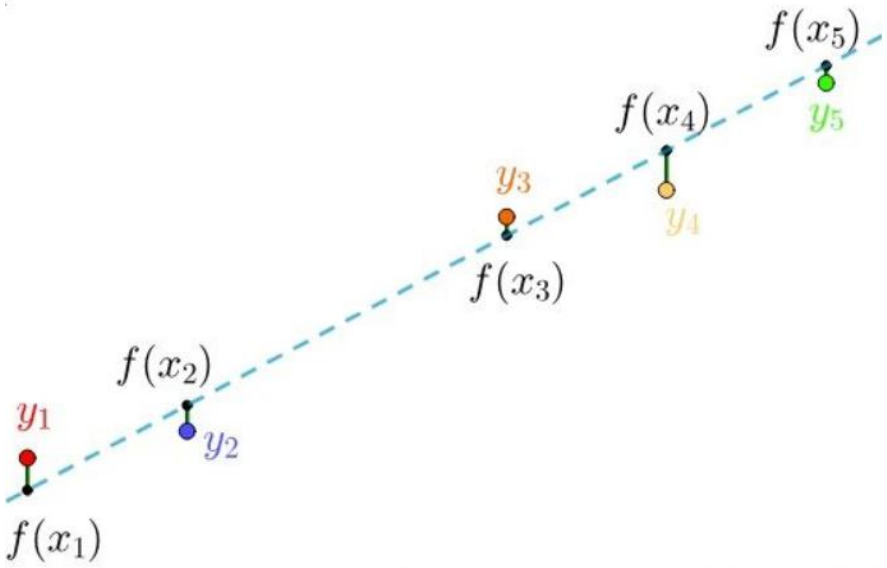
为了简化计算，令

$$\begin{aligned}l(\boldsymbol{\theta})&=\log L(\boldsymbol{\theta})\\&=m\log\frac{1}{\sqrt{2\pi}}+\sum_{i=0}^{m}-\frac{(y^{(i)}-\hat{y}^{(i)}|_{\boldsymbol{\theta}})^2}{2\sigma}\end{aligned}$$

要让$L(\boldsymbol{\theta})$最大，即需让$l(\boldsymbol{\theta})$最大，即让$\sum_{i=0}^{m}(y^{(i)}-\hat{y}^{(i)}|_{\boldsymbol{\theta}})^2$取到最小值。

实例 2.线性回归：

上面我们讨论了最简单的情况：测量值为常数，下面我们再接着看复杂一点的情况，即函数关系为线性关系。



若假定所满足的线性关系为：f(x)=ax+b，根据最小二乘法思想，误差平方为：

$$S_{\epsilon^2} = \sum (f(x_i) - y_i)^2 = \sum (ax_i + b - y_i)^2$$

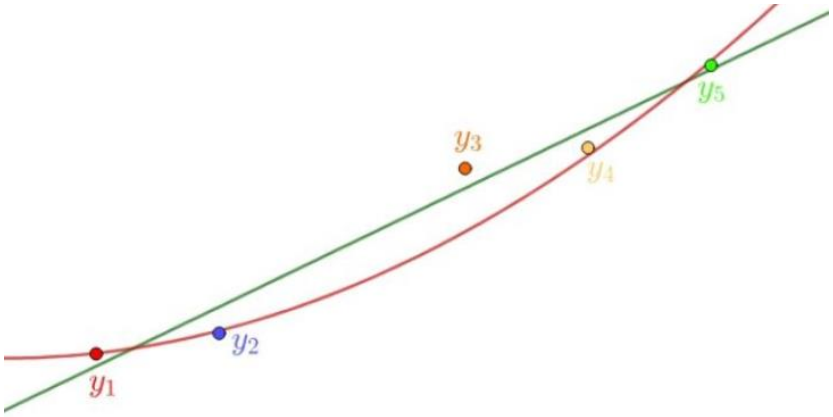
根据多元微积分知识，上式在其对 a、b 导数为 0 时取最小值，即：

$$\begin{cases} \frac{\partial}{\partial a} S_{\epsilon^2} = 2\sum (ax_i + b - y_i)x_i = 0 \\ \\ \frac{\partial}{\partial b} S_{\epsilon^2} = 2\sum (ax_i + b - y_i) = 0 \end{cases}$$

求解上述方程组即可得到 a、b 取值：

$$\begin{cases} a = \dfrac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \\ b = \bar{y} - a\bar{x} \end{cases}$$

曲线拟合讨论：



同样的，我们还可以使用最小二乘法拟合二次甚至是更高次的曲线，选择不同的函数关系 f(x)，我们都能够通过最小二乘法得到相应的曲线。这样对于同一组数据，实际上的拟合结果与我们选择的拟合关系式是相关的，而这种拟合关系是我们必须在数据处理之前事先通过物理模型或者经验得到的，这使得我们在做数据拟合之前会很多事前的工作或经验，这一点可以使用后续会讲到的神经网络来解决。

Python 进行最小二乘法拟合：

使用 scipy.optimize 包里面的 leastsq 函数。

函数介绍页：

https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.leastsq.html

格式：

scipy.optimize.leastsq(func, x0, args=(), Dfun=None, full_output=0, col_deriv=0, ftol=1.49012e-08, xtol=1.49012e-08, gtol=0.0, maxfev=0, epsfcn=None, factor=100, diag=None)

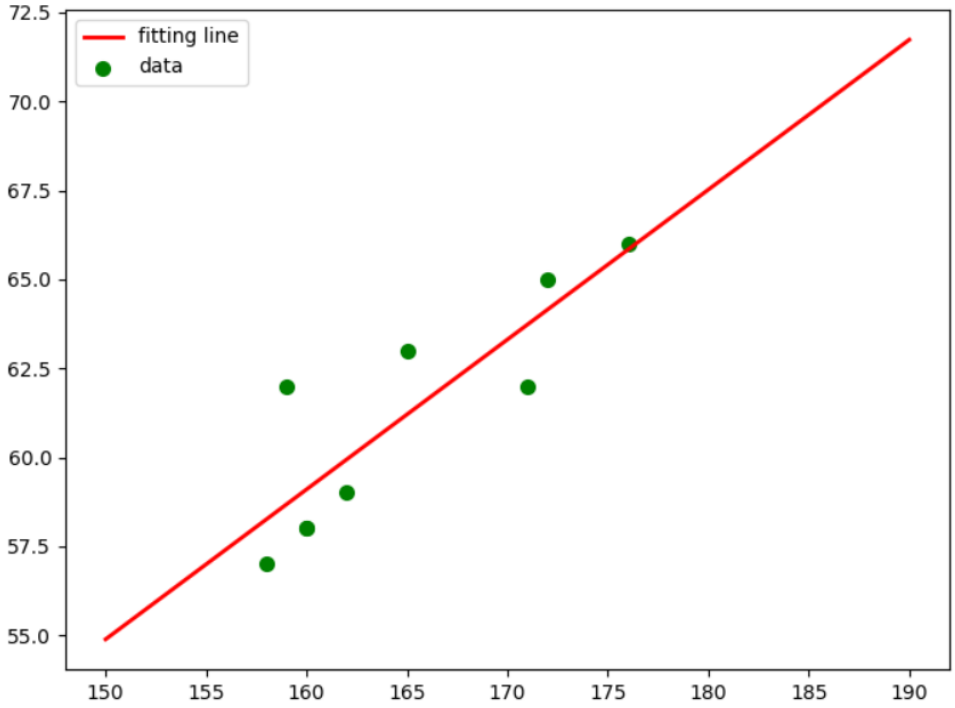
一般我们只要指定前三个参数就可以了:func 是我们自己定义的一个计算误差的函数，x0 是计算的初始参数值，args 是指定 func 的其他参数

以一个线性拟合的简单情况为例：

```
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
from scipy.optimize import leastsq
##样本数据(Xi,Yi)，需要转换成数组(列表)形式
Xi=np.array([160,165,158,172,159,176,160,162,171])
Yi=np.array([58,63,57,65,62,66,58,59,62])
def func(p,x):##需要拟合的函数func
    k,b=p
    return k*x+b
def error(p,x,y):##偏差函数：x,y都是列表:这里的x,y更上面的Xi,Yi中是一一对应的
    return func(p,x)-y
p0=[1,20]#k,b的初始值
#把error函数中除了p0以外的参数打包到args中(使用要求)
Para=leastsq(error,p0,args=(Xi,Yi))
#读取结果
k,b=Para[0]
print("k=",k,"b=",b)
#画样本点
plt.figure(figsize=(8,6)) ##指定图像比例： 8：6
plt.scatter(Xi,Yi,color="green",label="样本数据",linewidth=2)
#画拟合直线
x=np.linspace(150,190,100) ##在150-190直接画100个连续点
y=k*x+b ##函数式
plt.plot(x,y,color="red",label="拟合直线",linewidth=2)
plt.legend() #绘制图例
plt.show()
```

其结果显示为：

```
k= 0.42116973935029356 b= -8.28830260654588
```



使用一些其他常见软件进行拟合：

实际上，在实际工作或者实验中如果遇到需要进行拟合的情况，除了使用上面的方式或者自己编写最小二乘法进行计算之外，我们还可以使用一些成熟的软件中提供的相关功能：

1. Excel
2. OriginPro